VAX CONFIGURATION UTILITY (IVV000)

USER'S GUIDE

3545 Harbor Boulevard
Costa Mesa, California 92626
(714) 662-5600 TWX 910-595-2521

VX9950905 Rev D
February, 1986

Printed in U.S.A.

TABLE OF CONTENTS

Section		Page
ONE	**GENERAL DESCRIPTION**	
1.1	**INTRODUCTION**	1-1
1.2	**PRODUCT OVERVIEW**	1-1
1.3	**DISTRIBUTION MEDIA**	1-1
1.4	**COMPATIBILITY**	1-2
1.4.1	HARDWARE	1-2
1.4.2	SOFTWARE	1-2
1.5	**RELATED DOCUMENTATION**	1-2
TWO	**OPERATION**	
2.1	**OVERVIEW**	2-1
2.2	**LOAD AND START PROCEDURES**	2-1
2.3	**SAMPLE OUTPUT**	2-1
2.3.1	VAX-11/730	2-1
2.3.2	VAX-11/750	2-2
2.3.3	VAX-11/780	2-4

BLANK

Section 1
GENERAL DESCRIPTION

1.1 INTRODUCTION

This manual is designed to serve as a guide for those using the Emulex VAX configuration utility, IVV000, on Digital Equipment Corporation (DEC) VAX-11 and VAX-8600 computers. IVV000 is designed to run under the Emulex VAX Monitor, EVM, and the Emulex VAX 8600 Monitor, EVM 8600.

This diagnostic autosizer program is designed for use by qualified installers of Emulex equipment, and thus it assumes that the user has some knowledge of hardware configurations, VAX architecture and terminology, and interpretation of error messages and device register contents.

This document contains two main sections:

Section 1	General Description: This section contains an overview of IVV000, including its functions, distribution media, hardware and software compatibility, and related documentation.
Section 2	Operation: Describes operation ot IVV000, including load and start procedures and sample dialog.

1.2 PRODUCT OVERVIEW

The purpose of the IVV000 diagnostic is to print a map of all populated and unpopulated input/output devices, so that the operator can determine the hardware configuration ot the system.

1.3 DISTRIBUTION MEDIA

The following table lists and describes distribution media for IVV000 and other Emulex VAX and MicroVAX diagnostic software.

Emulex P/N	Description
VX9960405	TU58 cassette for VAX-11/750
VX9960505	Eight-inch floppy diskette for VAX-11/780
VX9960910	9-track mag tape for VAX-8600
VX9960704	5.25-inch floppy diskette for MicroVAX

1.4 COMPATIBILITY

1.4.1 HARDWARE

IVV000 is an autosizer for all VAX buses. It is compatible with DEC VAX 11/730, 11/750, 11/780, and 8600, and MicroVAX I and II computers.

1.4.2 SOFTWARE

IVV000 is designed to run with the Emulex VAX diagnostic monitor, EVM, with EVM 8600, and with the Emulex MicroVAX diagnostic monitor, MicroEVM. For information regarding these diagnostic monitors, see the user's guides referenced in subsection 1.5.

1.5 RELATED DOCUMENTATION

Title: Emulex VAX Monitor (EVM) User's Guide
Publication Number: VX9950901
Publisher: Emulex Corporation
3545 Harbor Blvd.
Costa Mesa, CA 92626
(714) 662-5600 TWX 910-595-2521

Title: Emulex VAX-8600 Monitor (EVM 8600) User's Guide
Publication Number: VX9950924
Publisher: Emulex Corporation
3545 Harbor Blvd.
Costa Mesa, CA 92626
(714) 662-5600 TWX 910-595-2521

2.1 OVERVIEW

This section describes IVV000 load and start procedures, and explains sample program output.

User input appears in **bold** type, in order to distinguish it from IVV000 or EVM output. The symbol **<return>** represents the carriage return key.

2.2 LOAD AND START PROCEDURES

The procedure used to invoke EVM varies from one VAX system to another. For a description of EVM bootstrapping procedures, see the EVM user's guide (reference given in subsection 1.5).

After the EVM> prompt has appeared on the screen, you can obtain information regarding EVM commands by typing:

```
EVM>HELP<return>
```

To load and start IVV000, type the following. (The default filename extension is .EXE.)

```
EVM>LOAD IVV000<return>
EVM>START<return>
```

The START command may be abbreviated **ST.** Because IVV000 is an autosizer, it requires no SET CONFIGURATION statement in order to run.

2.3 SAMPLE OUTPUT

For definitions of technical terms and acronyms used in this subsection, please refer to the VAX Hardware Handbook (reference given in subsection 1.5).

Statements that appear on the right-hand side of the page, preceded by exclamation points, are explanatory comments rather than IVV000 output. They are provided here in order to clarify the significance of the output. In subsection 2.3.2, for example, note that there is only one attention summary (AS) register for all eight possible devices. Thus the device 0 AS register responds for drives 2 through 7, even though the devices do not exist.

IVV000 sample output is shown only for VAX-11 systems. Output for MicroVAX systems is similar in appearance.

2.3.1 VAX-11/730

The sample output presented in this subsection refers to a VAX-11/730 with the following configuration:

- One UDA50 disk drive at address 772150_8
- One DMF32 communications multiplexer at address 760340_8
- One TU80 tape drive at address 772520_8

```
EVM>LOAD IVV000<return>
EVM>START<return>

Emulex Config Utility  REV 1.0  dd-mmm-yyyy Time

TEST # 1  Configuration Utility
  dd-mmm-yyyy Time


--------------CONFIG--------------

CPU ID = 00000003, VAX-11/730

*** UBA0:
*** UBA_BASE_ADR = 00FC0000
Unpopulated .... 00760000 - 00760336
Populated ...... 00760340 - 00760376
Unpopulated .... 00760400 - 00772146
Populated ...... 00772150 - 00772152
Unpopulated .... 00772154 - 00772516
Populated ...... 00772520 - 00772522
Unpopulated .... 00772524 - 00777776

-----------------------------------

SUMMARY REPORT:
  TOTAL # ERRORS = 0 (0 SYSTEM, 0 DEVICE, 0 HARD, 0 SOFT)
  dd-mmm-yyyy Time
```

2.3.2 VAX-11/750

The sample output presented in this subsection refers to a VAX-11/750 with the following configuration:

- Two RH750 disk subsystems at addresses $F28000_{16}$ and $F2A000_{16}$
- One UNIBUS adapter at address $FC0000_{16}$
- Two DMF32 communications multiplexers at addresses 760340_8 and 760400_8
- One TU80 tape drive at address 772520_8

In the following sample output, the address range $00F28400_{16}$ through $00F284FC_{16}$ represents address space for 32 external registers for each device, 0 and 1, from byte offset 400-4FC. Similarly, the address range $00F2A400_{16}$ through $00F2A47C_{16}$ represents space for 32 external registers for device 0 in this sample configuration. Even though space for 32 external registers is available to the firmware for each device, the actual valid address range used for registers is device dependent, as described in each controller technical manual. Any attempt to access a register outside the range specified for a given device causes the illegal register bit to be set in the error register.

```
     EVM>LOAD IVV000<return>
     EVM>START<return>

Emulex Config Utility   REV 1.0   dd-mmm-yyyy Time

TEST # 1  Configuration Utility
  dd-mmm-yyyy Time

--------------CONFIG--------------

CPU ID = 00000002, VAX-11/750

*** UBA0:
*** UBA_BASE_ADR = 00FC0000
Unpopulated .... 00760000 - 00760336
Populated ...... 00760340 - 00760436
Unpopulated .... 00760440 - 00772516
Populated ...... 00772520 - 00772522
Unpopulated .... 00772524 - 00777776

*** MBA0:
*** MBA_BASE_ADR = 00F28000
Populated ...... 00F28400 - 00F284FC          ! All drive 0 and 1
Unpopulated .... 00F28500 - 00F2850C            registers respond
Populated ...... 00F28510 - 00F28510          ! Drive 2 AS register
Unpopulated .... 00F28514 - 00F2858C
Populated ...... 00F28590 - 00F28590          ! Drive 3 AS register
Unpopulated .... 00F28594 - 00F2860C
Populated ...... 00F28610 - 00F28610          ! Drive 4 AS register
Unpopulated .... 00F28614 - 00F2868C
Populated ...... 00F28690 - 00F28690          ! Drive 5 AS register
Unpopulated .... 00F28694 - 00F2870C
Populated ...... 00F28710 - 00F28710          ! Drive 6 AS register
Unpopulated .... 00F28714 - 00F2878C
Populated ...... 00F28790 - 00F28790          ! Drive 7 AS register
Unpopulated .... 00F28794 - 00F287FC
```

```
*** MBA1:
*** MBA_BASE_ADR = 00F2A000
Populated ..... 00F2A400 - 00F2A47C          ! All drive 0 registers
Unpopulated .... 00F2A480 - 00F2A7FC           respond
```

2.3.3 VAX-11/780

The sample output given below refers to a VAX-11/780 with the following configuration:

- One memory controller (TR 1)
- One UNIBUS adapter (TR 3) at address 20100000_{16}
- One DMF32 communications multiplexer at address 760340_{8}
- One TS11 tape subsystem at address 772520_{8}
- Two RH780 disk subsystems (TR 8 and TR 9) at addresses 20010000_{16} and 20012000_{16} respectively

```
EVM>LOAD IVV000<return>
EVM>START<return>

Emulex Config Utility  REV 1.0  dd-mmm-yyyy Time

TEST # 1  Configuration Utility
  dd-mmm-yyyy Time

--------------CONFIG--------------

CPU ID = 00000001, VAX-11/780

TR = 1

NEXUS other than UBA/MBA; TR = 1

TR = 3

*** UBA0:
*** UBA_BASE_ADR = 20100000
Unpopulated .... 00760000 - 00760336
Populated ..... 00760340 - 00760376
Unpopulated .... 00760400 - 00772516
Populated ..... 00772520 - 00772522
Unpopulated .... 00772524 - 00777776

TR = 8

*** MBA0:
*** MBA_BASE_ADR = 20010000
Populated ..... 20010400 - 2001047C          ! All drive 0 registers
Unpopulated .... 20010480 - 2001048C           respond
Populated ..... 20010490 - 20010490          ! Drive 1 AS register
Unpopulated .... 20010494 - 2001050C
```

```
Populated ..... 20010510 - 20010510          ! Drive 2 AS register
Unpopulated .... 20010514 - 2001058C
Populated ..... 20010590 - 20010590          ! Drive 3 AS register
Unpopulated .... 20010594 - 2001060C
Populated ..... 20010610 - 20010610          ! Drive 4 AS register
Unpopulated .... 20010614 - 2001068C
Populated ..... 20010690 - 20010690          ! Drive 5 AS register
Unpopulated .... 20010694 - 2001070C
Populated ..... 20010710 - 20010710          ! Drive 6 AS register
Unpopulated .... 20010714 - 2001078C
Populated ..... 20010790 - 20010790          ! Drive 7 AS register
Unpopulated .... 20010794 - 200107FC

TR = 9

*** MBA1:
*** MBA_BASE_ADR = 20012000
Populated ..... 20012400 - 2001247C          ! All drive 0 registers
Unpopulated .... 20012480 - 2001248C            respond
Populated ..... 20012490 - 20012490          ! Drive 1 AS register
Unpopulated .... 20012494 - 2001250C
Populated ..... 20012510 - 20012510          ! Drive 2 AS register
Unpopulated .... 20012514 - 2001258C
Populated ..... 20012590 - 20012590          ! Drive 3 AS register
Unpopulated .... 20012594 - 2001260C
Populated ..... 20012610 - 20012610          ! Drive 4 AS register
Unpopulated .... 20012614 - 2001268C
Populated ..... 20012690 - 20012690          ! Drive 5 AS register
Unpopulated .... 20012694 - 2001270C
Populated ..... 20012710 - 20012710          ! Drive 6 AS register
Unpopulated .... 20012714 - 2001278C
Populated ..... 20012790 - 20012790          ! Drive 7 AS register
Unpopulated .... 20012794 - 200127FC

---------------------------------------

SUMMARY REPORT:
  TOTAL # ERRORS = 0 (0 SYSTEM, 0 DEVICE, 0 HARD, 0 SOFT)
  dd-mmm-yyyy Time
```

BLANK

MASSBUS DISK FORMATTER (FMD000)

USER'S GUIDE

EMULEX
3545 Harbor Boulevard
Costa Mesa, California 92626
(714) 662-5600 TWX 910-595-2521

VX9950902 Rev E
April, 1986

Printed in U.S.A.

TABLE OF CONTENTS

Section		Page
ONE	**GENERAL DESCRIPTION**	
1.1	**INTRODUCTION**	1-1
1.2	**PRODUCT OVERVIEW**	1-1
1.3	**DISTRIBUTION MEDIA**	1-1
1.4	**COMPATIBILITY**	1-2
1.4.1	HARDWARE	1-2
1.4.2	SOFTWARE	1-2
1.5	**RELATED DOCUMENTATION**	1-2
TWO	**OPERATION**	
2.1	**OVERVIEW**	2-1
2.2	**LOAD AND START PROCEDURES**	2-1
2.2.1	LOAD PROCEDURE	2-1
2.2.2	SAMPLE CONFIGURE STATEMENT FOR VAX-11/750	2-1
2.2.3	SAMPLE CONFIGURE STATEMENT FOR VAX-11/780	2-2
2.2.4	START PROCEDURE	2-2
2.3	**TEST SECTIONS**	2-3
2.3.1	TEST 1: FORVER (FORMAT AND VERIFY)	2-3
2.3.2	TEST 2: FORMAT	2-3
2.3.3	TEST 3: VERIFY	2-4
2.3.4	TEST 4: UPDATE	2-5
2.3.5	TEST 5: READALL	2-6
2.3.6	TEST 6: DISPLAY	2-6
2.4	**SAMPLE DIALOG**	2-6

BLANK

Section 1
GENERAL DESCRIPTION

1.1 INTRODUCTION

This manual is designed to serve as a guide for those using the Emulex MASSBUS Disk Formatter, FMD000, on Digital Equipment Corporation (DEC) VAX-11 computers. FMD000 is designed to run under the Emulex VAX Monitor, EVM.

This utility is designed for use by qualified installers of Emulex equipment, and thus it assumes that the user has some knowledge of hardware configurations, VAX architecture and terminology, and interpretation of error messages and device register contents.

This document contains two main sections, the contents of which are described briefly below.

Section 1	General Description: This section contains an overview of FMD000, including its functions, hardware and software compatibility, distribution media, and related documentation.
Section 2	Operation: Describes operation of FMD000, including load and start procedures, test sections, and sample dialog.

1.2 PRODUCT OVERVIEW

The FMD000 MASSBUS Disk Formatter is used to prepare new disks for use and/or to maintain the integrity of disks already in use.

1.3 DISTRIBUTION MEDIA

The following table lists and describes distribution media for FMD000 and other Emulex VAX diagnostic software.

Emulex P/N	Description
VX9960406	TU58 cassette for VAX-11/750
VX9960506	Eight-inch floppy diskette for VAX-11/780
VX9960910	9-track mag tape for VAX-8600

1.4 COMPATIBILITY

1.4.1 HARDWARE

FMD000 is compatible with DEC VAX-11/750 and 11/780 computers. It tests the following Emulex disk controllers:

SC750/B1/B2/B3
SC758/B1
SC780/B1/B2/B3
SC788/B1
SC7000/B1

1.4.2 SOFTWARE

FMD000 is designed to run with the Emulex VAX Monitor, EVM. For information on EVM, see the EVM User's Guide, referenced in subsection 1.5.

1.5 RELATED DOCUMENTATION

Documents listed in this subsection can be ordered from the following address:

Emulex Corporation
3545 Harbor Blvd.
Costa Mesa, CA 92626
(714) 662-5600 TWX 910-595-2521

Title:	Emulex VAX Monitor (EVM) User's Guide
Publication Number:	VX9950901
Title:	VAX Configuration Utility (IVV000) User's Guide
Publication Number:	VX9950905
Title:	SC7000/B1 (RM03/RM05/RM80 Compatible) Disk Controller Technical Manual
Publication Number:	SC7551004
Title:	SC788/B1 (RM03/RM05/RM80 Compatible) Disk Controller Technical Manual
Publication Number:	SC7851003
Title:	SC780/B2 (RP04/RP05/RP06 Compatible) Disk Controller Technical Manual
Publication Number:	SC7851002
Title:	SC780/B1 SC780/B2 (RM03/RM05/RM80 Compatible) Disk Controller Technical Manual
Publication Number:	SC7851001

Related Documentation

Title:	SC785/B1 (RM03/RM05/RM80 Compatible) Disk Controller Technical Manual
Publication Number:	SC7551003

BLANK

Section 2
OPERATION

2.1 OVERVIEW

This section describes FMD000 load and start procedures, presents sample configure statements, defines the six tests available with FMD000, and presents sample output.

User input appears in **bold** type, in order to distinquish it from program output. The symbol **<return>** represents the carriage return key.

2.2 LOAD AND START PROCEDURES

The procedure used to invoke EVM varies from one VAX system to another. For a description of EVM bootstrapping procedures, see the Emulex VAX Monitor (EVM) User's Guide (reference given in subsection 1.5).

2.2.1 LOAD PROCEDURE

After the EVM> prompt has appeared on the screen, type the following. (The default filename extension is .EXE.)

```
EVM>LOAD FMD000<return>
```

The LOAD statement is followed by a SET CONFIGURATION statement, the content of which depends upon the configuration of the VAX system being used. Sample configure statements are presented in the following subsections.

2.2.2 SAMPLE CONFIGURE STATEMENT FOR VAX-11/750

The following statement refers to a VAX-11/750 with an Emulex SC7xx/B3 controller, one Fujitsu M2351 at drive 0, and MASSBUS address RH1: (base address F2A000):

```
EVM>SET CONFIG/ADAPTER:1/DRIVE:0<return>
```

Acceptable values for ADAPTER are 0 through 2:

```
ADAPTER 0     RH0. F28000
ADAPTER 1     RH1. F2A000
ADAPTER 2     RH2. F2C000
```

2.2.3 SAMPLE CONFIGURE STATEMENT FOR VAX-11/780

The following statement refers to a VAX-11/780 with an Emulex SC7xx/B3 controller. one Fujitsu M2351 at drive 0. and MASSBUS address RH1 (which corresponds to TR 9):

```
EVM>SET CONFIG/TR:9/DRIVE:0<return>
```

In the preceding statement. acceptable values for TR are 1 through 15. but the range 8 through 11 is the convention:

```
TR 8     RH0. 20010000
TR 9     RH1. 20012000
TR 10    RH2. 20014000
TR 11    RH3. 20016000
```

2.2.4 START PROCEDURE

After typing the LOAD statement and the appropriate SET CONFIGURATION statement. type:

```
EVM>START/TEST:n/PASSES:x<return>
```

where **n** represents a test number in the range 1 through 6. and **x** represents the desired number of passes in the range 0 through 100.

For FMD000. a test number must be specified; see subsection 2 3 for descriptions of the six available tests. If /TEST is omitted. EVM attempts to run all six tests. starting with test 1. This will work for many EVM compatible diagnostic programs; FMD000. however. requires that all tests except test 1 be called specifically in order to initialize properly. Therefore. if no test number is specified. test 1 runs but test 2 fails with the following error message:

```
ILLEGAL TEST SEQUENCE
```

Although FMD000 tests cannot be chained by default. you can run tests 2 and 3 (format and verify) as a unit by selecting test 1. which combines these two options. See subsection 2.3.1 for details.

The qualifier **/PASSES** is optional and needs to be specified only if multiple passes are required. The default value for **/PASSES** is 1; acceptable values are 0 through 100. where 0 signifies an infinite number of passes.

2.3 TEST SECTIONS

The six available diagnostic tests are described in the following subsections. After FMD000 has been loaded, information regarding a given test section can also be obtained by typing the following:

```
EVM>HELP/DIAG<return>
```

2.3.1 TEST 1: FORVER (FORMAT AND VERIFY)

The pack format and verify section optionally initializes the bad/skip sector file, and then performs the format operation followed by the verify test. For detailed descriptions of these two tests, see subsections 2.3.2 and 2.3.3 respectively. This section is provided in order to allow the format and verify sections to be executed as a unit, because test sections cannot otherwise be chained.

During test setup, before any test sections have executed, the operator is prompted to select a 16-bit mask, as explained in subsection 2.3.3.

2.3.2 TEST 2: FORMAT

The pack format utility formats the disk; it is not a test of the data area. It writes sector headers and data fields, using the firmware format feature. Data fields are written with all zeros. The bad/skip sector files are initialized if requested (see subsection 2.3.1).

The operator is given the option to initialize any existing bad sector data, if either Test 1 or Test 2 has been selected and if the pack already contains some bad sector information:

- A Y (yes) response to this option causes all bad (and skipped) sector entries to be deleted.
- A N (no) response to this prompt causes the program to preserve all pre-existing bad sector information. The data is saved in memory so that it can be rewritten on the pack after the formatting process is complete.

NOTE

If the FMD000 disk maintenance utility program is aborted before the format operation has been completed, **all** pre-existing bad sector data is lost.

2.3.3 TEST 3: VERIFY

The verify test performs a surface scan and adds any errors encountered to the bad/skip sector file. The updating of the bad/skip sector file occurs after the program has finished writing each pattern.

Table 2-1 lists the 16-word verification data patterns that are used to verify the media. When the program prompts you to select verification patterns. input a 16-bit mask in which each bit corresponds to a single pattern in Table 2-1: bit 0 = pattern 0. bit 1 = pattern 1. and so on.

Patterns 0 through 3 are selected by default (mask 17 octal). because this combination of patterns usually finds most bad spots on the disk. The time required for the verify test increases with the number of patterns selected.

Table 2-1. Verification Data Patterns (Octal)

Pat-0	Pat-1	Pat-2	Pat-3	Pat-4	Pat-5	Pat-6	Pat-7
000000	165555	133333	177777	000001	177776	000000	000000
000000	133333	165555	177777	000003	177774	000000	010421
000000	165555	133333	177777	000007	177770	000000	021042
000000	133333	165555	177777	000017	177760	177777	031463
000000	165555	133333	177777	000037	177740	177777	042104
000000	133333	165555	177777	000077	177700	177777	052525
000000	165555	133333	177777	000177	177600	000000	063146
000000	133333	165555	177777	000377	177400	000000	073567
000000	165555	133333	177777	000777	177000	177777	104210
000000	133333	165555	177777	001777	176000	177777	114631
000000	165555	133333	177777	003777	174000	000000	125252
000000	133333	165555	177777	007777	170000	177777	135673
000000	165555	133333	177777	017777	160000	000000	146314
000000	133333	165555	177777	037777	140000	177777	156735
000000	165555	133333	177777	077777	100000	177777	167356
000000	133333	165555	177777	177777	000000	000000	177777
Pat-8	**Pat-9**	**Pat-10**	**Pat-11**	**Pat-12**	**Pat-13**	**Pat-14**	**Pat-15**
052525	007417	026455	077577	000001	177776	172666	077777
052525	007417	026455	077577	000002	177775	155555	137777
052525	007417	026455	077577	000004	177773	172666	157777
125252	170360	151322	077577	000010	177767	155555	167777
125252	170360	151322	077577	000020	177757	172666	173777
125252	170360	151322	077577	000040	177737	155555	175777
052525	007417	026455	077577	000100	177677	172666	176777
052525	007417	026455	077577	000200	177577	155555	177377
125252	170360	151322	077577	000400	177377	172666	177577
125252	170360	151322	077577	001000	176777	155555	177677
052525	007417	026455	077577	002000	175777	172666	177737
125252	170360	151322	077577	004000	173777	155555	177757
052525	007417	026455	077577	010000	167777	172666	177767
125252	170360	151322	077577	020000	157777	155555	177773
125252	170360	151322	077577	040000	137777	172666	177775
052525	007417	026455	077577	100000	077777	155555	177776

2.3.4 TEST 4: UPDATE

The manual update section allows the operator to manually update the bad/skip sector file by adding or deleting sectors.

CAUTION

The update test section must be used with caution, or the integrity of the volume may be jeopardized.

2.3.5 TEST 5: READALL

The readall test reads the entire disk and reports all errors encountered, not including known bad spots. It does not destroy data on the disk, and can be run with the disk write-protected.

2 3.6 TEST 6: DISPLAY

The display section reads and displays all of the recorded bad/skip sector files from the selected disk.

2.4 SAMPLE DIALOG

This subsection explains operation of FMD000 and presents sample user dialog. In this example, the operator has requested only test 1 (format and verify). The number of passes is not specified, so the test is performed once.

The general format of FMD000 prompts is as follows:

```
Query [DEC - min.max.(def)]>>>
```

where DEC signifies decimal radix; min and max are the minimum and maximum acceptable values respectively; and def is the default response that the program uses if you enter only **<return>**. Other abbreviations that appear in prompts are OCT (octal radix), Y (yes), and N (no).

FMD000 asks you to input the date, if the drive is an RM80. Then it displays the date and asks you to verify that it is correct. Next, it warns that the FORVER operation will destroy data, and asks if you wish to continue.

If you indicate that you want to continue, the program asks you to enter a 16-bit mask, as explained in subsection 2 3.3 of this document.

Finally, the program asks you to input the pack serial number. You may select any decimal number in the specified range.

```
EVM>START/TEST:1<return>

Emulex VAX-MASSBUS disk maintenance utility  REV X.1  DD-MM-YYYY
Time

Unit 0
RM80  (842 Cylinders,  20 Tracks,  48 Sectors)
```

```
TEST # 1

          ** PACK FORMAT AND VERIFY SECTION **

dd-mmm-yyyy Time                    Please enter today's date -

    Month [DEC - 1.12]>>> 10<return>

    Day [DEC - 1.30]>>> 30<return>

    Year [DEC - 84. 1999]>>> 1984<return>

Current date is 10/30/84

Is this correct [Y.N.(Y)]>>> <return>

Reading bad sector files > Done

The FORVER section will write on the pack and may destroy data.

Do you want to continue [Y.N.(N)]>>> Y<return>

    Select patterns to be used during verify [OCT -
    1.177777.(17)]>>> <return>

    Enter pack serial No.  [DEC - 1.2147483467.(1)]>>> 123<return>

    Beginning format on unit 0 at 0:0:0
    Format completed at 0:4:9
    Rewriting bad sector data.
    Beginning verification with pattern 3 at Time
    Current No. of bad spots is x
    Pattern 3 finished at Time
    Beginning verification with pattern 2 at Time
    Current No. of bad spots is x
    Pattern 2 finished at Time
    Beginning verification with pattern 1 at Time
    Current No. of bad spots is x
    Pattern 1 finished at Time
    Beginning verification with pattern 0 at Time
    Current No. of bad spots is x
    Pattern 0 finished at Time

    x new bad spots were found.


    SUMMARY REPORT:
      TOTAL # ERRORS = 0 (0 SYSTEM. 0 DEVICE. 0 HARD. 0 SOFT)
      dd-mmm-yyyy Time
```

BLANK